UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/729,771	12/08/2003	Erik de Groot	I20 04992US	3404

128 7590 08/02/2006

HONEYWELL INTERNATIONAL INC.
101 COLUMBIA ROAD
P O BOX 2245
MORRISTOWN, NJ 07962-2245

EXAMINER	
PHAM, THAI V	

ART UNIT	PAPER NUMBER
2194	

DATE MAILED: 08/02/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/729,771	DE GROOT ET AL.	
	Examiner	Art Unit	
	Thai Van Pham	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 December 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *08 December 2003* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

This is the initial office action based on the application filed on July 19, 2006. Claims 1 – 37 are currently pending and have been considered below.

Drawings

1. Figure 13 is objected to because of the following informality: Mislabeling of "RETURN STATUS". The specification refers to "RETURN STATUS" as item 1304 of the figure; however, it is labeled as 1309.

Claim Objections

2. Claims 15 and 27 are objected to because of the following informalities: lack of antecedent bases.

-- Claim 15: The claim recites "said current state transition" (on page 23) which is not previously identified in the claim.

-- Claim 27: The claim recites "said server" which is not previously identified in the parent claim (Claim 26) or the claim itself. Based on the disclosure of the source control system, The Examiner assumes that the server being recited here is simply a redundant server that backs up the source control system and automatically takes over the system when the primary server fails.

Appropriate correction is required.

Examiner's Note

3. The technical terminologies used in the claim language listed below are non-conventional in the art of software development. The scope of a claim is thus limited to

the definitions of these terminologies as they are explicitly defined in the disclosure of the application.

-- A fallback state: the state that an object is placed in when it is checked-in to the source control system, after it has been checked-out from the source control system.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1 – 3, 7 – 8, 11, 13, 16 – 18, 20 – 21, 23, 26, 28 – 30, 32 – 33, and 36 – 37 are rejected under 35 U.S.C. 102(b) as being anticipated by **Fiszman** (6,115,646).

-- Claim 1: **Fiszman** discloses a method for enforcing a life cycle process in a source control system, comprising:

- receiving a user-defined life cycle process having a plurality of states, each state having attributes (i.e., activities and their individual attributes; Fig. 1, page 5 line 7 – page 6 line 3);
- receiving user-defined state transitions between said plurality of states (i.e., transition properties of activities; Fig. 12, page 15 line 38 – page 16 line 41);
- providing a change state function for changing a current state associated with an object to a next state associated with said object, said change state function verifying compliance with said user-defined state transitions (page 6 lines 20 – 50);

• and providing version control for said object in said source control system (Fig. 17, page 18 lines 44 – 63).

-- Claim 2: **Fiszman** discloses the method according to claim 1 where the version control inherently must comprise providing a check-in function and providing a check-out function.

-- Claim 3: **Fiszman** discloses the method according to claim 1 and further discloses the attributes include a fallback state (i.e., activities defined within a particular process; Fig. 9, page 14 line 43 – page 15 line 13).

-- Claim 7: **Fiszman** discloses the method according to claim 1, where the object is a control strategy for a process control system (page 6 lines 30 – 34).

-- Claim 8: **Fiszman** discloses the method according to claim 7, where the attributes include whether said control strategy is loadable to a controller (page 7 line 49 – page 8 line 6).

-- Claim 11: **Fiszman** discloses a computer readable medium having executable instructions stored thereon to perform a method of determining permissions for actions with an object based on a state of said object, said method comprising:

• receiving a request to perform an action with said object (i.e., execution of an a particular activity by a processing node or a role; page 5 lines 21 – 28);

• determining whether said object has ever been checked-in to a source control system (i.e., determining existence of activity associated with version control; Figs. 11 and 17, page 15 lines 30 –37);

• determining whether said object is currently checked-in (i.e., determining existence of activity associated with version control; Figs. 11 and 17, page 15 lines 30 – 37);

• retrieving a definition of said state of said object (i.e., displaying existing activities definitions; Fig. 11, page 15 lines 30 –37);

• determining from said definition whether said action is permissible in said state (i.e., attributes defined for an activity; Fig. 13, page 16 line 42 – page 17 line 8); and

• providing a permission status (i.e., feedback and output; Fig. 1, page 5 lines 35 – 40).

-- Claim 13: **Fiszman** discloses a computer readable medium having executable instructions stored thereon to perform a method of validating a state transition, said method comprising:

• determining whether a next state in a state transition request from a user is allowed from a current state in said state transition request based on user-defined transition restrictions (Fig. 16B, page 17 line 59 – page 18 line 36);

• determining whether said user has permission to make said state transition based on user-defined transition restrictions (Fig. 16B, page 17 line 59 – page 18 line 36); and

• providing a state transition status (Fig. 1, page 5 lines 35 – 40).

-- Claim 16: **Fiszman** discloses a computer readable medium having executable instructions stored thereon to perform a method of determining a new state for an object version upon check-in, said method comprising:

• determining whether said object is being checked-in for a first time (i.e., an inherent property of version control; Fig. 17);

• retrieving a first fallback state for a first pre-defined state, if said object is being checked-in for said first time (i.e., activities defined within a particular process; Fig. 9, page 14 line 43 – page 15 line 13); and

• providing said first fallback state, if said object is being checked-in for said first time (i.e., activities defined within a particular process; Fig. 9, page 14 line 43 – page 15 line 13).

-- Claim 17: **Fiszman** discloses the computer readable medium according to claim 16, comprising:

• retrieving a current state for a current version of said object, if said object is not being checked-in for said first time (i.e., retrieving an existing activity definition; Figs. 13 –14, page 16 line 42 – page 17 line 8);

• retrieving a current fallback state for said current state of said object, if said object is not being checked-in for said first time (i.e., retrieving an existing activity definition; Figs. 13 –14, page 16 line 42 – page 17 line 8); and

• providing said current fallback state, if said object is not being checked-in for said first time (i.e., retrieving an existing activity definition; Figs. 13 –14, page 16 line 42 – page 17 line 8).

-- Claim 18: **Fiszman** discloses a computer readable medium having executable instructions stored thereon to perform a method of processing the addition of a state, said method comprising:

• receiving a definition of a new state from a user, said definition including a name and a fallback state (i.e., creating activities in a process definition; Figs. 9 – 11, page 14 line 43 – page 15 line 37);

• determining whether said name is unique among existing state definitions (i.e., listing activities in a process definition; Figs. 9 – 11, page 14 line 43 – page 15 line 37);

• validating said fallback state (i.e., distinctive activities; Fig. 4); and

• adding said definition to a source control system, only if said name is unique and said fallback state is valid (i.e., an inherent property of version control; Fig. 17).

-- Claim 20: **Fiszman** discloses the computer readable medium according to claim 18 and further discloses that the method further comprising determining whether said user has a privilege to edit said definition; and wherein said adding said definition to said source control system is performed on an additional condition of whether said user has said privilege (i.e., access authorization; page 5 lines 53 – 56).

-- Claim 21: **Fiszman** discloses a computer readable medium having executable instructions stored thereon to perform a method of processing the modification of a state, said method comprising:

- receiving a modified definition of a state from a user, said modified definition including a name and a fallback state (i.e., editing activities in a process definition; Figs. 9 – 11, page 14 line 43 – page 15 line 37);
- determining whether said name is unique among existing state definitions (i.e., listing activities in a process definition; Figs. 9 – 11, page 14 line 43 – page 15 line 37);
- validating said fallback state (i.e., distinctive activities; Fig. 4); and
- updating said modified definition in a source control system, only if said name is unique and said fallback state is valid (i.e., an inherent property of version control; Fig. 17).

-- Claim 23: **Fiszman** discloses the computer readable medium according to claim 21, further comprising: determining whether said user has a privilege to edit said definition; and wherein said updating said modified definition in said source control system is performed on an additional condition of whether said user has said privilege (i.e., access authorization; page 5 lines 53 – 56).

-- Claim 26: **Fiszman** discloses a source control system for a process control system, comprising:

- a processor (an inherent property in a computer system);

• a life cycle process component executable on said processor to enforce compliance with user-defined life cycle states (Fig. 3, page 8 lines 11 – 33);

• a version control component executable on said processor to associate a version number with each object (Fig. 17, page 18 lines 37 – 63); and

• a controller in communication with said processor via a network to be loaded with said objects to provide process control for a plurality of devices (page 7 line 49 – page 8 line 6).

-- Claim 28: **Fiszman** discloses the system according to claim 26, further comprising: a state configuration component executable on said processor to receive state information from a user for each state (i.e., defining activity attributes; Figs. 9 –13).

-- Claim 29: **Fiszman** discloses the system according to claim 28, wherein said state information includes a state name and an indication of whether load to controller is allowed from that state (Figs. 13 and 14, page 16 line 42 – page 15 line 37; page 7 line 49 – page 8 line 6).

-- Claim 30: **Fiszman** discloses the system according to claim 28, wherein said state information includes a fallback state (i.e., activities defined within a particular process; Fig. 9, page 14 line 43 – page 15 line 13).

-- Claim 32: **Fiszman** discloses the system according to claim 28, wherein said state configuration component provides editing functions for said state information (Fig. 13, page 16 lines 42 – 45).

-- Claim 33: **Fiszman** discloses the system according to claim 26, further comprising: a state transition component executable on said processor to receive state transition configuration requirements from a user (Fig. 12, page 15 line 38 – page 16 line 41).

-- Claim 36: **Fiszman** discloses the system according to claim 26, wherein said version control component provides check-in and check-out functions (i.e., an inherent property of version control; Fig. 17).

-- Claim 37: **Fiszman** discloses the system according to claim 26, further comprising: a change qualification state component to process a qualification state transition request from a user (Fig. 12, page 15 line 38 – page 16 line 41).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 24 – 25, and 27 are rejected under 35 U.S.C. 103(a) as being obvious over **Fiszman** (6,115,646).

-- Claim 24: **Fiszman** discloses a computer readable medium having executable instructions stored thereon to perform a method of processing the addition and modification of a state, the method comprising:

• receiving a modified definition of a state from a user, said modified definition including a name and a fallback state (i.e., editing activities in a process definition; Figs. 9 – 11, page 14 line 43 – page 15 line 37). Although **Fiszman** does not explicitly disclose that the method comprises receiving a request to delete a state definition for said state from a user, **Fiszman**'s disclosure of the modified definition of an activity above would have made it obvious to one of ordinary skills in the art of software development at the time the invention was made to add to the method of **Fiszman** a method of processing a deletion of an activity when the activity becomes obsolete in a process or when it is mistakenly created by the user. The deletion of the activity can be performed in the activity graph of the process definition in Fig. 9 where the addition and modification take place;

• determining whether said name is unique among existing state definitions and validating said fallback state (i.e., listing activities in a process definition; Figs. 9 – 11, page 14 line 43 – page 15 line 37). Although **Fiszman** but does not explicitly disclose that the method comprises determining whether said state definition is referenced by any other state definition in a source control system and determining whether any objects in said source control system have a current state equal to said state, **Fiszman**'s disclosure of defining/modifying the attributes of activities (Figs. 9 and 13) shows how transition properties of an activity are determined in relation to the existence of activities to and from which the transitions take place under certain conditions. Therefore, prior to an activity deletion, it is necessary to examine the dependencies of the activity to make proper to modification to the process as a whole. Thus, it would

have been obvious to one of ordinary skills in the art of software development at the time the invention was made to further allow the method of processing the deletion of a state in **Fiszman** to determine whether the state is referenced by any other state in the source control system; and

• updating said modified definition in a source control system, only if said name is unique and said fallback state is valid (i.e., an inherent property of version control; Fig. 17). Although **Fiszman** does not further explicitly disclose that the method deleting said state definition from said source control system, only if said state definition is not referenced by any other state definition in said source control system and no objects in said source control system have said current state equal to said state, it would have been obvious that once an activity of the method is determined to be obsolete and proper modification to the process is made to purge the process of any reference made to the obsolete activity, the activity can be removed safely. Thus, it would have been obvious to one of ordinary skills in the art of software development at the time the invention was made to further allow the method of processing the deletion of a state in **Fiszman** to delete the state definition when it is obsolete in the source control system.

-- Claim 25: **Fiszman** discloses the computer readable medium according to claim 24, further comprising: determining whether said user has a privilege to delete said definition; and wherein said deleting said state definition from said source control system is performed on an additional condition of whether said user has said privilege (i.e., access authorization; page 5 lines 53 – 56).

-- Claim 27: **Fiszman** discloses the system according to claim 26, further comprising: another processor to back-up said server. Official Notice is taken that it is old and well known in the art of software development that data on a computer can always be backed up over a network. Thus, the process control system can be backed up by a secondary server, which inherently must contain a processor. Therefore, it would have been obvious to one of ordinary skills in the art of software development at the time the invention was made to further allow the method of processing the deletion of a state in **Fiszman** to have a secondary back-up server containing a processor.

8. Claim 15 is rejected under 35 U.S.C. 103(a) as being obvious over **Wisnosky** (2003/0190593).

-- Claim 15: **Wisnosky** discloses a computer readable medium having executable instructions stored thereon to perform a method of validating a state transition, said method comprising:

- determining whether said current state transition in a state transition request for an object from a user requires an electronic signature based on user-defined transition restrictions (i.e., different areas of the system employ various security settings to enable a certain user to gain different access levels to a particular area of the system; Fig. 6D, [0081 – 0082]);
- but **Wisnosky** does not explicitly disclose that the method determines whether a previous state transition for said object required a previous electronic signature, if said current state transition requires a current electronic signature; allowing said current

state transition only if said previous electronic signature is different than said current electronic signature. Official Notice is taken that it is old and well known in business accounting practices that the process for generating a purchase order always requires at least two different signatures for approval. In particular, a person who makes the determination of what needs to be purchased generates a purchase order explaining the need and detailing the price and specification of the items to be purchased; he then signs it to authenticate the order (i.e., purchase order generation state). Subsequently, a manager or a person having the authority to approve the purchase order must sign it (i.e., purchase order approval state) before a check is issued my another person having the authority to make the financial decision (i.e., payment state). It is clear in this example that the signature required to go from "purchase order approval state" to "payment state" must be different than that required to go from "purchase order generation state" to "purchase order approval state". Thus, it would have been obvious to one of ordinary skills in the art of software development at the time the invention was made to further allow the method of **Wisnosky** to compare previous and current electronic signatures to validate the current state transition;

- providing a validation status (Fig. 6D, [0081 – 0082]).

9. Claims 4 – 6, 9 – 10, 12, 14, 19, 22, 31, and 34 – 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Fiszman** (6,115,646) and in view of **Wisnosky** (2003/0190593).

-- Claim 4: **Fiszman** discloses the method according to claim 1 but does not explicitly disclose the method further comprising: receiving user-defined security for said user-defined state transitions. **Wisnosky** discloses a system implementing a method for automatically generating individual transition plans according to user input statistical data; the system includes security setting that permits access to certain areas of the plan by authorized users. Only authorized user is granted access to the system implementing the method of **Wisnosky** (Fig. 5, [0076]); and furthermore, different areas of the system employ various security settings to enable a certain user to gain different access levels to a particular area of the system (Fig. 6D, [0081 – 0082]). Specifically, a particular user group (e.g., managers) is given all read, write, and delete permissions to an enterprise area; whereas a different user group (e.g., workers) is given only read and write permissions to the same area. The states of an individual transition plan change as data is created or removed by a user with proper authorization. The system implementing the method of **Fiszman** requires only user authorization for gaining system access (page 5 lines 53 – 56); and it would be more secured and robust if it further requires specific user security settings as disclosed **Wisnosky** so that only certain groups of users are given permissions to modify and/or delete certain processes or their corresponding activities. In other words, specific user security settings would provide the method of **Fiszman** with the ability to enable user-defined security for individual activity transitions and/or process transitions as a whole. Thus, it would have been obvious to one of ordinary skills in the art of software development at the time the

invention was made to further provide the method of **Fiszman** with a user-defined security for state transitions in addition to access authorization.

-- Claim 5: **Fiszman** and **Wisnosky** disclose the method according to claim 4 and **Wisnosky** further discloses the user-defined security includes electronic signatures (Fig. 5, [0076]).

-- Claim 6: **Fiszman** and **Wisnosky** disclose the method according to claim 4 and **Wisnosky** further discloses the user-defined security includes which users have permission to make which state transitions (i.e., read – write – delete; Fig. 6D, [0081 – 0082]).

-- Claim 9: **Fiszman** and **Wisnosky** disclose the method according to claim 1 and **Wisnosky** further discloses that receiving said user-defined life cycle process having said plurality of states, each state having attributes is performed through a user interface having an editable table, said table having state names as rows and attributes as columns and having cells indicating values for said attributes (Fig. 6D, [0081 – 0082]).

-- Claim 10: : **Fiszman** and **Wisnosky** disclose the method according to claim 6 and **Wisnosky** further discloses that receiving user-defined state transitions between said plurality of states is performed through a user interface having an editable table, said table having state names as rows and column and having cells indicating which users

have permission to make which state transitions (i.e., read – write – delete for each specific area; Fig. 6D, [0081 – 0082]).

-- Claim 12: **Fiszman** discloses a computer readable medium having executable instructions stored thereon to perform a method of validating state transitions, said method comprising:

- receiving a request to make a state transition for an object from a user (page 5 lines 21 – 28);
- determining whether said object is checked-in (an inherent property of version control; Fig. 11, page 15 lines 30 –37);
- however, **Fiszman** does not explicitly disclose that the method determines whether said user has permission to make said state transition based on a user-defined state transition model. **Wisnosky** discloses a system implementing a method for automatically generating individual transition plans according to user input statistical data; the system includes security setting that permits access to certain areas of the plan by authorized users. Only authorized user is granted access to the system implementing the method of **Wisnosky** (Fig. 5, [0076]); and furthermore, different areas of the system employ various security settings to enable a certain user to gain different access levels to a particular area of the system (Fig. 6D, [0081 – 0082]). Specifically, a particular user group (e.g., managers) is given all read, write, and delete permissions to an enterprise area; whereas a different user group (e.g., workers) is given only read and write permissions to the same area. The states of an individual transition plan change as data is created or removed by a user with proper authorization. The system

implementing the method of **Fiszman** requires only user authorization for gaining system access (page 5 lines 53 – 56); and it would be more secured and robust if it further requires specific user security settings as disclosed **Wisnosky** so that only certain groups of users are given permissions to modify and/or delete certain processes or their corresponding activities. In other words, specific user security settings would provide the method of **Fiszman** with the ability to enable user-defined security for individual activity transitions and/or process transitions as a whole. Thus, it would have been obvious to one of ordinary skills in the art of software development at the time the invention was made to further provide the method of **Fiszman** with the ability to determe whether the user has permission to make said state transition based on a user-defined state transition model;

- permitting said state transition, if said user has permission. If a user satisfies the security requirements, it is obvious for the method to permit the state transition; and
- providing a state transition status (Fig. 1, page 5 lines 35 – 40).

-- Claim 14: **Fiszman** discloses the computer readable medium according to claim 13 but does not explicitly disclose the method further comprising determining whether said state transition has a restricted signing requirement and, if so, verifying that said restricted signing requirement is met. **Wisnosky** discloses a system implementing a method for automatically generating individual transition plans according to user input statistical data; the system includes security setting that permits access to certain areas of the plan by authorized users. Only authorized user is granted access to the system implementing the method of **Wisnosky** (Fig. 5, [0076]); and furthermore, different areas

of the system employ various security settings to enable a certain user to gain different access levels to a particular area of the system (Fig. 6D, [0081 – 0082]). Specifically, a particular user group (e.g., managers) is.given all read, write, and delete permissions to an enterprise area; whereas a different user group (e.g., workers) is given only read and write permissions to the same area. The states of an individual transition plan change as data is created or removed by a user with proper authorization. The system implementing the method of **Fiszman** requires only user authorization for gaining system access (page 5 lines 53 – 56); and it would be more secured and robust if it further requires specific user security settings as disclosed **Wisnosky** so that only certain groups of users are given permissions to modify and/or delete certain processes or their corresponding activities. In other words, specific user security settings would provide the method of **Fiszman** with the ability to enable user-defined security for individual activity transitions and/or process transitions as a whole. Thus, it would have been obvious to one of ordinary skills in the art of software development at the time the invention was made to further enable security for user-defined state transition model in the method of **Fiszman** , and furthermore, provide the model with the ability to determine if a state transition has a restricted signing requirement as well as to verify if the restricted signing requirement is met.

-- Claim 19: **Fiszman** discloses the computer readable medium according to claim 18 but does not explicitly disclose that the definition includes a restricted signing requirement and further comprising validating said restricted signing requirement; and wherein said adding said definition to said source control system is performed on an

additional condition of whether said restricted signing requirement is valid. **Wisnosky** discloses a system implementing a method for automatically generating individual transition plans according to user input statistical data; the system includes security setting that permits access to certain areas of the plan by authorized users. Only authorized user is granted access to the system implementing the method of **Wisnosky** (Fig. 5, [0076]); and furthermore, different areas of the system employ various security settings to enable a certain user to gain different access levels to a particular area of the system (Fig. 6D, [0081 – 0082]). Specifically, a particular user group (e.g., managers) is given all read, write, and delete permissions to an enterprise area; whereas a different user group (e.g., workers) is given only read and write permissions to the same area. The states of an individual transition plan change as data is created or removed by a user with proper authorization. The system implementing the method of **Fiszman** requires only user authorization for gaining system access (page 5 lines 53 – 56); and it would be more secured and robust if it further requires specific user security settings as disclosed **Wisnosky** so that only certain groups of users are given permissions to modify and/or delete certain processes or their corresponding activities. In other words, specific user security settings would provide the method of **Fiszman** with the ability to enable user-defined security for individual activity transitions and/or process transitions as a whole. Thus, it would have been obvious to one of ordinary skills in the art of software development at the time the invention was made to further include a restricted signing requirement in the state definition in the method of **Fiszman** , and furthermore, provide the model with the ability to validate the restricted signing requirement and to

restrict adding the state definition to the source control system on an additional condition of whether said restricted signing requirement is valid.

-- Claim 22: **Fiszman** discloses the computer readable medium according to claim 21 but does not explicitly disclose that the definition includes a restricted signing requirement and further comprising: validating said restricted signing requirement; and wherein said updating said modified definition in said source control system is performed on an additional condition of whether said restricted signing requirement is valid. **Wisnosky** discloses a system implementing a method for automatically generating individual transition plans according to user input statistical data; the system includes security setting that permits access to certain areas of the plan by authorized users. Only authorized user is granted access to the system implementing the method of **Wisnosky** (Fig. 5, [0076]); and furthermore, different areas of the system employ various security settings to enable a certain user to gain different access levels to a particular area of the system (Fig. 6D, [0081 – 0082]). Specifically, a particular user group (e.g., managers) is given all read, write, and delete permissions to an enterprise area; whereas a different user group (e.g., workers) is given only read and write permissions to the same area. The states of an individual transition plan change as data is created or removed by a user with proper authorization. The system implementing the method of **Fiszman** requires only user authorization for gaining system access (page 5 lines 53 – 56); and it would be more secured and robust if it further requires specific user security settings as disclosed **Wisnosky** so that only certain groups of users are given permissions to modify and/or delete certain processes

or their corresponding activities. In other words, specific user security settings would provide the method of **Fiszman** with the ability to enable user-defined security for individual activity transitions and/or process transitions as a whole. Thus, it would have been obvious to one of ordinary skills in the art of software development at the time the invention was made to further include a restricted signing requirement in the state definition in the method of **Fiszman** , and furthermore, provide the model with the ability to validate the restricted signing requirement and to restrict updating the state definition to the source control system on an additional condition of whether said restricted signing requirement is valid.

-- Claim 31: **Fiszman** discloses the system according to claim 28 but does not explicitly disclose that the state information includes an indication of whether restricted signing is needed. **Wisnosky** discloses a system implementing a method for automatically generating individual transition plans according to user input statistical data; the system includes security setting that permits access to certain areas of the plan by authorized users. Only authorized user is granted access to the system implementing the method of **Wisnosky** (Fig. 5, [0076]); and furthermore, different areas of the system employ various security settings to enable a certain user to gain different access levels to a particular area of the system (Fig. 6D, [0081 – 0082]). Specifically, a particular user group (e.g., managers) is given all read, write, and delete permissions to an enterprise area; whereas a different user group (e.g., workers) is given only read and write permissions to the same area. The states of an individual transition plan change as data is created or removed by a user with proper authorization. The system

implementing the method of **Fiszman** requires only user authorization for gaining system access (page 5 lines 53 – 56); and it would be more secured and robust if it further requires specific user security settings as disclosed **Wisnosky** so that only certain groups of users are given permissions to modify and/or delete certain processes or their corresponding activities. In other words, specific user security settings would provide the method of **Fiszman** with the ability to enable user-defined security for individual activity transitions and/or process transitions as a whole. Thus, it would have been obvious to one of ordinary skills in the art of software development at the time the invention was made to further add to the state definition of the method of **Fiszman** an optional restricted signing requirement which facilitates the state information to include an indication of whether restricted signing is needed.

-- Claim 34 and 35: **Fiszman** discloses the system according to claim 33 but does not explicitly disclose that the state transition configuration requirements include which users have permission to make particular state transitions as well as an indication of whether an electronic signature is needed to make particular state transitions. **Wisnosky** discloses a system implementing a method for automatically generating individual transition plans according to user input statistical data; the system includes security setting that permits access to certain areas of the plan by authorized users. Only authorized user is granted access to the system implementing the method of **Wisnosky** (Fig. 5, [0076]); and furthermore, different areas of the system employ various security settings to enable a certain user to gain different access levels to a particular area of the system (Fig. 6D, [0081 – 0082]). Specifically, a particular user

group (e.g., managers) is given all read, write, and delete permissions to an enterprise area; whereas a different user group (e.g., workers) is given only read and write permissions to the same area. The states of an individual transition plan change as data is created or removed by a user with proper authorization. The system implementing the method of **Fiszman** requires only user authorization for gaining system access (page 5 lines 53 – 56); and it would be more secured and robust if it further requires specific user security settings as disclosed **Wisnosky** so that only certain groups of users are given permissions to modify and/or delete certain processes or their corresponding activities. In other words, specific user security settings would provide the method of **Fiszman** with the ability to enable user-defined security for individual activity transitions and/or process transitions as a whole. Thus, it would have been obvious to one of ordinary skills in the art of software development at the time the invention was made to further add to the state transition configuration requirements of the method of **Fiszman** an optional restricted signing requirement which facilitates the state transition configuration requirements to include an indication of whether restricted signing is needed.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

-- **Nath** et al. (US 2005/0071658), PSS Systems, Inc.: a method and system for securing digital assets using process-driven security policies.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thai Van Pham whose telephone number is (571) 270-1064. The examiner can normally be reached on Monday - Thursday, 9am - 5pm EST. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, James Myhre can be reached on (571) 270-1065. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

TVP

TVP
7/19/2006

James Myhre
Supervisory Patent Examiner